



VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

Teemu Korkalainen

PARAMETRIOHJELMA

Forssan Metallityöt Oy

Tekniikka
2018

TIIVISTELMÄ

Tekijä	Teemu Korkalainen
Opinnäytetyön nimi	Parametriojelma
Vuosi	2018
Kieli	suomi
Sivumäärä	29 + 2 Liitettä
Ohjaaja	Pirjo Prosi

Päätavoitteena tässä opinnäytetyössä oli kehittää FM-oville ohjelma, jolla pystytään luomaan tuotantolaitteita ohjaavat tiedostot sekä ovien että karmien mitoista. Yrityksellä jo ennestään käytössä olevalla ohjelmalla voidaan luoda tiedot parametreista vain ovien mitoille. Karmien mitat kirjoitetaan ja ladataan käsin suoraan karmikoneen tietokoneelta.

Työ toteutettiin avoimen lähdekoodin Electron-kirjastolla, johon on sisäänrakennettu Chromium ja Node.js. Tämä mahdollistaa natiivien työpöytäsovelluksien kehittämisen kaikille työpöytäympäristöille nykypäivän Web-teknologioita käyttämällä. Työssä käytetään JavaScript-, CSS-, ja HTML-ohjelmointikieliä.

Asiakkaan vaatimuksina oli pystyä kehittämään selkeäkäyttöinen sovellus, jolla voidaan luoda tiedostot ovien ja karmien ominaisuuksista ja tiedoista. Näitä pitää voida myös muuttaa tarvittaessa.

Työn lopputuloksena saatiin uusittu ohjelma, parannellulla käyttäjäkokemuksella, joka täyttää vaaditut kriteerit.

ABSTRACT

Author	Teemu Korkalainen
Title	Parameter Software
Year	2018
Language	Finnish
Pages	29 + 2 Appendices
Name of Supervisor	Pirjo Prosi

The main purpose of this thesis was to develop a program that can create necessary parameters for the manufacture of solid fire doors and their frames for FM doors. The company's own program can only create parameters for the doors but not for the frames. The frames must be entered manually to a parameter file and then load it onto the machine from its computer.

The program was developed using an open source JavaScript library called Electron, which has integrated Chromium and Node.js. This makes it possible to develop native programs for Windows, Linux and Mac OS by using modern-day Web-technologies. The programming languages what was used for this program were JavaScript, CSS and HTML.

The client's minimum requirements were to create a user-friendly software that can create corresponding parameters in a file for both the doors and the frames. An additional requirement was to be able to change the data for the doors and the frames visible in the program. The minimum requirements were met, the software has upgraded user-experience and possibility to modify the visible data in the program.

KÄYTETYT TERMIT JA LYHENTEET

Electron	Avoin JavaScript- kehitysalusta
JavaScript	Web-ohjelmointikieli
CSS	Porrastettu tyyliarkki
HTML	Hypertekstin merkinäkieli
GIT	Versionhallintaohjelmisto
Node	JavaScript ajoympäristö
V8	JavaScript- moottori
Chromium	Avoin selainprojekti
Electron-json-storage	Avoin JavaScript- kirjasto
Electron-packager	Avoin JavaScript- kirjasto
Bootstrap	Avoin front-end- kirjasto
Bootstrap-toggle	Avoin lisäosa bootstrapille
FM-Ovet	Forssan Metallityöt
Salvagnini	Ovikone
Pivatic	Karmikone
API	Ohjelmointirajapinta

SISÄLLYS

TIIVISTELMÄ

ABSTRACT

KÄYTETYT TERMIT JA LYHENTEET

KUVIOLUETTELO

LIITELUETTELO

1	JOHDANTO	9
2	YRITYS	10
2.1	Tausta.....	10
2.2	Parametriojelma.....	10
3	TEKNOLOGIAT JA OHJELMISTOT.....	12
3.1	Electron	12
3.1.1	Chromium	12
3.1.2	Main	12
3.1.3	Renderer	13
3.2	Node.js.....	14
3.3	V8.....	14
3.4	IndexedDB.....	14
3.5	Electron-json-storage.....	15
3.6	Electron-packager.....	15
3.7	Git.....	15
3.8	Atom.....	16
3.9	Bootstrap	16
3.9.1	Bootstrap-toggle.....	17
3.10	jQuery-ui	17
4	PROJEKTIN VAIHEET.....	20

4.1	Aloitius	20
4.2	Suunnittelu.....	20
4.3	Toteutus	21
4.4	Testaus	26
5	YHTEENVETO	28
	LÄHDELUETTELO	29

LIITTEET

KUVIOLUETTELO

Kuvio 1 Vanha parametriojelma	10
Kuvio 2 Parametritiedosto	11
Kuvio 3 Main ja Renderer (Nokes, 2016)	13
Kuvio 4 JSON-tiedosto.....	14
Kuvio 5 Vaihtopainike, Salvagnini	17
Kuvio 6 Vaihtopainike, Pivatic	17
Kuvio 7 Karmit.....	18
Kuvio 8 Tiivistekarmit.....	19
Kuvio 9 Vakio tuotelista	22
Kuvio 10 Teksti-ikkunat ja valintaruudut	23
Kuvio 11 Pudotusvalikko.....	23
Kuvio 12 Tiedostojen luonti	24
Kuvio 13 Windows-komentorivitiedosto.....	25
Kuvio 14 Tekstitiedosto	25
Kuvio 15 Pivatic, lataus.....	25
Kuvio 16 Sovelluskansio	26
Kuvio 17 Pivatic, parametritiedosto	27
Kuvio 18 Uusi parametriojelma, Salvagnini	30
Kuvio 19 Uusi parametriojelma, Pivatic	31

LIITELUETTELO

Taulukko 1 Vaatimusluettelo

1 JOHDANTO

Tämä opinnäytetyö tehtiin Forssan metallitöille heidän vanhan ohjelman päivittämiseksi. Parametriohjelmalla luodaan parametritiedostoja metallilevykoneille, jonka mukaan koneet valmistavat palo- tai turvaovet. Parametri on arvo, joka välittyy käskynä koneelle, joko ihmisen syöttämänä tai toisen tietokoneen kautta. Parametritiedostot sisältävät monta riviä käskyjä, jotka kone tunnistaa ja valmistaa tuotteen niissä määrättyjen ominaisuuksien mukaisesti.

Aikaisemman version on luonut tuotantovastaava Jyrki Kivijärvi, käyttämällä Visual Basic for Applications -ohjelmointikieltä, joka toimii Excelin makro-ohjelmilla. Hän toimii myös yrityksen yhteyshenkilönä tähän opinnäytetyöhön liittyen. Ongelmana vanhalla ohjelmalla oli, että sillä voidaan luoda parametritiedostot vain toiselle koneista. Toiselle koneelle pitää kirjoittaa parametrit erikseen toiseen tiedostoon ja ladata koneelle sen tietokoneelta.

Opinnäytetyö tehtiin käyttämällä JavaScript-, HTML-, ja CSS -ohjelmointikieliä, sekä avoimen lähdekoodin Electron JavaScript-kirjastoa. Työssä käytetään myös IndexedDB:tä JSON- tiedostojen parsimiseen ja avointa electron-json-storage-kirjastoa, joka on Electronin virallisen yhteisön luoma työkalu. Työkalulla luodaan ja ladataan ohjelman käyttämät JSON- tiedostot, jossa säilytetään tuotetiedot ja parametrit. Työkalu kuuluu MIT-lisenssin alle, joka mahdollistaa sen käytön myös ”kaupallisissa suljetun lähdekoodin ohjelmistoissa.” - (Wikipedia, 2017). Projektissa käytetään myös saman yhteisön luomaa ”electron-packager”-kirjastoa testisovelluksien sekä tuotantosovelluksen luomisen helpottamiseksi.

2 YRITYS

2.1 Tausta

Forssan Metallityöt on vuonna 1960 perustettu palo- ja turvaovien valmistaja, joka palvelee rakentajia, talotehtaita ja jälleenmyyjiä. Yrityksellä on pitkälle automatisoitu valmistus, joka tarjoaa vakio- ja räätälöityjä ratkaisuja asiakkailleen. (Forssan Metallityöt, 2017). Yrityksen liikevaihto vuonna 2016 oli 8,9 miljoonaa euroa ja tilikauden tulos oli 362 000 euroa. (Finder, 2017)

2.2 Parametriohtelma

Forssan Metallitöillä on kaksi tuotantokonetta, Pivatic ja Salvagini. Parametriohtelma on yrityksessä kehitetty ohjelma, joka auttaa heitä tuotannossa luomalla parametritiedostoja tuotantokoneille. Ohjelma toimii Microsoft Excel-ympäristössä makro-ohjelmilla, jotka on kirjoitettu Visual Basic-ohjelmointikielellä. (Kuvio 1.)

Kuvio 1 Vanha parametriohtelma

Ohjelmasta valitaan vakiotuote, joka tallennetaan parametritiedostona kansioon tietyn polun alle. (Kuvio 2.) Tästä kansioista valitaan Salvagninin ohjelmalla

haluttu tiedosto, joka käännetään ja ladataan tuotantokoneelle. Kääntämisvaiheessa Salvagninin ohjelma tarkastaa tiedostossa annetut parametrit ja luo esikatselun tuotteesta, josta voidaan vielä tarkistaa, että tuote on oikeanlainen.

```
COD:" KANSI.S4 "  
BEX:  
PAR:ovi =( 1 )  
PAR:tyyppi =( 1 )  
PAR:purku =( 1 )  
PAR:tiivistek =( 0 )  
PAR:lampo =( 0 )  
PAR:merkki =( 0 )  
PAR:teksti = "123|KANSI|R9x21_OIK"  
PAR:eimagn =( 0 )  
PAR:korkeap =( 0 )  
PAR:omp =( 7.80 )  
PAR:mat ="DC01"  
PAR:matpituus =( 2145 )  
PAR:matleveys =( 1035 )  
PAR:paksuus =( 1 )  
PAR:trimauto =( 0 )  
PAR:profilh =( 0 )  
PAR:leveysl =( 778 )  
PAR:pituusl =( 2003 )  
PAR:kylkiluukpl =( 0 )  
PAR:katisyyssp =( 2 )  
PAR:keskifm =( 1 )  
PAR:saranasiirto =( 100 )  
PAR:lovi =( 5.1 )  
PAR:saranakeski =( 0 )  
PAR:lukko =( 3 )
```

Kuvio 2 Parametritiedosto

3 TEKNOLOGIAT JA OHJELMISTOT

3.1 Electron

Electron on yrityksen nimeltä GitHub kehittämä avoimen lähdekoodin JavaScript -kirjasto, jolla voi kehittää natiiveja työpöytäsovelluksia Windowsille, Linuxille sekä Mac OS:lle. Kirjaston kehitys alkoi 2013, jolle GitHub rakensi heidän oman tekstieditorinsa, Atomin. Molempien lähdekoodi muutettiin avoimiksi keväällä 2014. (Electron, 2017) Electronin käyttöönotto on yksinkertaista, se voidaan asentaa pakettinhallintaohjelmilla npm tai yarn, tai ladata ja kääntää lähdekooditiedostot. Asennus luo neljä päätiedostoa, joita Electron tarvitsee toimiakseen, main.js, renderer.js, index.html ja package.json. Tiedostojen nimillä ei ole merkitystä, kunhan viittaukset ovat oikein package.json- tiedostossa.

Electronin vanhemmille versioille ei ole tällä hetkellä pitkäaikaista tukea, vaan aikaisemmassa versiossa voi pysyä niin kauan kuin kehittäjä näkee tarpeelliseksi. Uusi versio julkaistaan, kun kehitysalustan rajapintaa päivitetään, alustaan tehdään korjauksia, tai Node ja/tai Chromium julkaistaan päivityksiä. (Electron, 2017)

3.1.1 Chromium

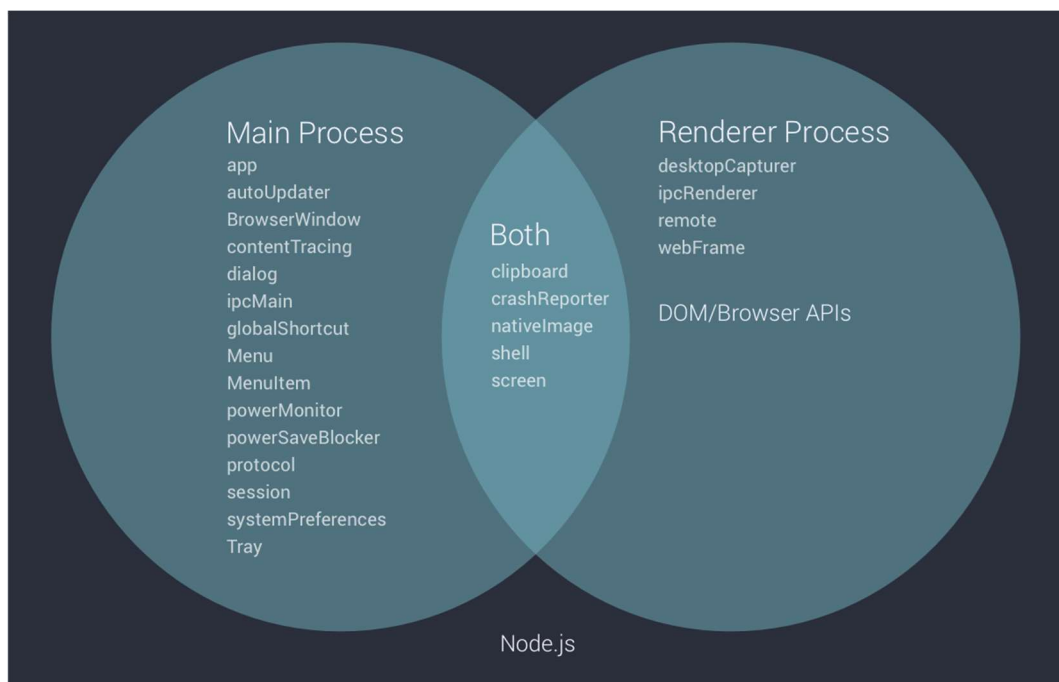
Chromium on avoimen lähdekoodin selainprojekti, jolle esim. Google on rakentanut oman Google Chrome -selaimen. (Google, 2017) Jotta Electronia olisi helpompi päivittää, se käyttää Chromiumin renderöintikirjastoa sen sijaan, että alustassa olisi koko Chromium- projekti paketoituna. (Electron, 2017)

3.1.2 Main

”Main” -prosessi käsittelee Electronin pääprosessiin kuuluvat metodit, esim. selainikkunan toiminnot ja erilaiset sovellustapahtumat. Siihen kuuluu selainikkunan lisäksi useita muita metodeja. (**Kuvio 3.**) Main- prosessissa tapahtuvat ohjelman ydintoiminnot, logiikka, luodaan käyttöikkuna ja käsitellään mitä sen avaamisen/sulkemisen aikana/jälkeen tapahtuu ja määritellään mitä HTML- tiedostoa käytetään ulkoasuna.

3.1.3 Renderer

”Renderer”- prosessi käsittelee selainkäyttöikkunaan kuuluvat metodit, esim. kaikki DOM, selain -ja node -rajapinnat. (**Kuvio 3.**) Siellä luodaan käyttöikkunan logiikka, toiminnallisuus, ja määrätään mitä tapahtuu esim. napin painalluksella. Renderöinti -ja pääprosessit voivat myös kommunikoida keskenään ”remote”- metodilla, joka mahdollistaa pääprosessin metodien käyttämisen renderöintiprosessissa ja renderöintiprosessin metodien käyttämisen pääprosessissa.



Kuvio 3 Main ja Renderer (Nokes, 2016)

3.2 Node.js

Node.js on alustariippumaton Chromen V8 JavaScript -moottorilla rakennettu serverityökalu, joka mahdollistaa JavaScriptin ajamisen serverillä. Noden paketinhallintaohjelma, npm, on maailman suurin ekosysteemi avoimen koodin kirjastoille. (Node.js Foundation, 2017). Integroidun node -kirjaston avulla Electron voi käyttää metodeja, jotka tavallisesti tapahtuvat serverillä, esim. tiedostojen ja kansioden luonti, niiden avaaminen, poistaminen, kirjoittaminen ja korvaaminen. Jotta Electronilla voi tehdä uskottavia työpöytäsovelluksia, tämä on yksi tärkeimmistä elementeistä.

3.3 V8

V8 on avoin JavaScript -ajoympäristö, jota käytetään Googlen Chrome-selaimessa ja Node.js:ssä. V8:lla suoritetaan JavaScript -koodia C++ -sovelluksissa.

3.4 IndexedDB

IndexedDB on oliopohjainen rajapinta suurien määrien datan hallitsemiseen selaintasolla. (Mozilla, 2017) Samaan tapaan kuin kirjastossa haetaan kirjoja aakkoskirjaimella, indexedDB:llä voidaan tallentaa ja hakea dataa, jolle on määrätty hakuavain. Tässä työssä IndexedDB:n tallennetaan hakuavaimet, joita käytetään tuotevakioiden hakemiseen JSON- tietokannasta objektin id:llä. Tässä tapauksessa tietokannalla viitataan käyttäjän tietokoneella paikallisesti luotua JSON-tyyppistä tiedostoa. Vakiot luodaan automaattisesti ohjelman käynnistyessä ensimmäisen kerran. (Kuvio 4.)

```
{
  "vedin": [
    {"id":0,"malli":"Ei vedintä"},
    {"id":1,"malli":"Primo inoxi 137-25/300"},
    {"id":2,"malli":"Primo 131/150 R"},
    {"id":3,"malli":"Primo 131/200 R"},
    {"id":4,"malli":"FM 131/200 R"}],
  "telki": [
    {"id":0,"malli":"Ei telkiaukkoa"},
    {"id":1,"malli":"Telkiaukko"},
    {"id":2,"malli":"MKTP"}],
}
```

Kuvio 4 JSON-tiedosto

3.5 Electron-json-storage

Projektissa käytetään Electronin virallisen yhteisön luomaa electron-json-storage-ratkaisua tietojen luomisen, tuomisen ja päivittämisen helpottamiseksi. Tällä luodaan ohjelman ensimmäisen käynnistyksen yhteydessä tietopankki, joka myös tarkistaa onko tiedostoja olemassa käyttäjän tietokoneella. Tämä myös mahdollistaa tietojen muokkaamisen ohjelman ollessa käytössä, joka vaatii vain ohjelman käyttö sivun päivittämisen käyttämällä ”Ctrl + R” -näppäinyhdistelmää tavallisen nettisivun tapaan. (Cruz Viotti, 2017)

3.6 Electron-packager

Electron-packager on komentorivityökalu, joka luo Electron sovelluksista sovelluskansion tuotantokäyttöä varten. Työkalulla voidaan paketoita sovellus Windows-(32-bit tai 64-bit), Mac OS X-sekä Linux-(x86/x86_64) -pohjaisille käyttöliittymille. Projektissa tätä on käytetty testisovelluksien sekä tuotantosovelluksen Windows -version luomiseen. Electron-packager paketoit ohjelman lähdekoodin, sekä luo ohjelmalle suoritustiedoston uudelleennimettyyn kansioon. (Electron-userland, 2017)

3.7 Git

Git on versionhallintaohjelmisto, jota käytetään tämän projektin versiointiin. Ohjelmisto luo projektista hakemistopuun, jossa se säilyttää historiatiedot jokaisesta siihen tehdystä muutoksesta. Jokin versionhallintaohjelma on tarpeellinen jokaisessa sovelluskehitysprojektissa, sillä jos jossain tapahtuu virhe, kehittäjän on helppo palata aiempaan versioon ja selvittää mistä virhe johtuu. Git luo puuhun yhden pääjuuren, johon luodaan kehityshaaroja muutoksia varten. Näin ollen juureen ei tehdä suoraan muutoksia, vaan ne tehdään haaraan, jotka yhdistetään juuren kanssa.

3.8 Atom

Atom on GitHubin kehittämä avoin tekstin muokkausohjelma, joka aloitettiin vuonna 2013 samaan aikaan Electronin kanssa. (Electron, 2017) Atomilla voidaan kirjoittaa ja muokata minkä tahansa tyyppistä ohjelmointikieltä. Siihen on myös integroitu perus GIT- komennot versionhallinnalle sekä yhteensopivuus GitHubin kanssa.

3.9 Bootstrap

Bootstrap-kirjaston kehittivät ohjelmistokehittäjät Mark Otto, [@mdo](#) ja Jacob, [@fat](#) Twitterissä vuonna 2010, joka tunnettiin silloin Twitter-pohjapiirroksena, ”Twitter blueprint” ennen julkistamista nykyisellä nimellään Bootstarp. (Twitter, 2017). Kirjasto on avoin front-end-kirjasto, joka toimii html-elementtien luokka-attribuuteilla. Elementille määrätään tietyn niminen luokka, joka hakee sille nimelle luodun CSS-tyylittelyn. Esimerkkinä alla oleva painike. Sille on määrätty luokka-attribuutiksi ”btn btn-primary”, jonka tyylittelyn kirjasto hakee sisäänrakennetusta ”btn btn-primary” CSS-luokasta. Bootstrap asennetaan noden npm-paketinhallintaohjelmalla tai lataamalla lähdekooditiedostot ja sisällyttämällä ne HTML-tiedostossa.

```
”<link href="assets/styles/bootstrap.min.css" rel="stylesheet"/>”
```

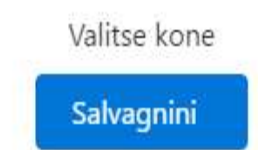
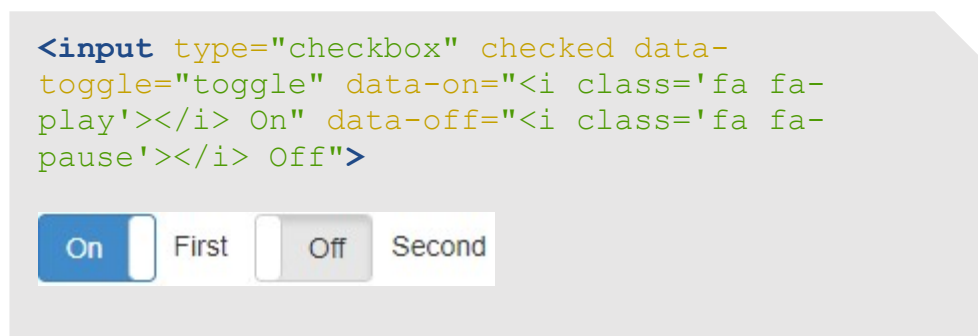
```
”<script>require('./js/bootstrap.min.js')</script>”
```

```
<button type="button" class="btn btn-  
primary"></button>
```



3.9.1 Bootstrap-toggle

Bootstrap-toggle on liitännäinen Bootstrap-kirjastoon, joka muuttaa valintaruudun vaihtopainikkeeksi. Esimerkkinä alla oleva kuva. Uudessa Parametriohjelmassa tätä käytetään sivun vaihtona Salvagninin ja Pivaticin käyttö sivujen välillä. (Kuviot 5-6.)



Kuvio 5 Vaihtopainike, Salvagnini



Kuvio 6 Vaihtopainike, Pivatic

3.10 JQuery-ui

JQuery-ui:ta käytetään Pivaticin käyttö sivun vaihtoikkunoissa. Vaihtoikkunoiden valinta määrää mitkä parametrit luodaan karmeja varten. (Kuviot 7-8.)

Karmit			Tiivistet		
Valio vaakakarmi			Valio pystykarmi		
Leveys	Kiinnitys		Pituus	Kiinnitys	
<input type="text" value="990"/>	<input type="text" value="kiinnitysreikä normaa"/>		<input type="text" value="2090"/>	<input type="text" value="kiinnitysreikä normaa"/>	
Korjaus	Ovityyppi		Lukkoh	Kynnys	
<input type="text" value="12"/>	<input type="text" value="yksittäinen ovi"/>		<input type="text" value="0"/>	<input type="text" value="ei"/>	
Käyntioven leveys	Kätisyys		Tlukkoh	Ovityyppi	
<input type="text" value="0"/>	<input type="text" value="oik"/>		<input type="text" value="0"/>	<input type="text" value="yksittäinen ovi: oik"/>	
	Mg Vaaka		Saranasiirto	Sarana	
	<input type="text" value="ei"/>		<input type="text" value="100"/>	<input type="text" value="FM23"/>	
	Salpa		Alaväli	Ylivienti	
	<input type="text" value="6/3, df3000, roca200,"/>		<input type="text" value="31"/>	<input type="text" value="ei"/>	
	Jätevaaka		Korjaus	Murtotappi	
	<input type="text" value="Aloitusjäte"/>		<input type="text" value="12"/>	<input type="text" value="ei"/>	
Kpl vaaka	Pöytävaaka	Stopparivaaka	Turvalukko	MG Pysty	
<input type="text" value="1"/>	<input type="text" value="3"/>	<input type="text" value="4"/>	<input type="text" value="ei"/>	<input type="text" value="ei"/>	
			Jätevaaka	Telki	
			<input type="text" value="Aloitusjäte"/>	<input type="text" value="vakio"/>	
			Kpl pysty	Pöytäpysty	Stopparipysty
			<input type="text" value="1"/>	<input type="text" value="3"/>	<input type="text" value="4"/>

Kuvio 7 Karmit

Karmit
Tiivisteet

Tiivistevaakakarmi

Leveys	Kiinnitys
<input type="text" value="990"/>	<input type="text" value="kiinnitysreikä normaa"/>
Korjaus	Ovityyppi
<input type="text" value="12"/>	<input type="text" value="yksittäinen ovi"/>
Käyntioven leveys	Kätsisyys
<input type="text" value="0"/>	<input type="text" value="oik"/>
	Mg Vaaka
	<input type="text" value="ei"/>
	Salpa
	<input type="text" value="6/3, df3000, roca200"/>
	Jätevaaka
	<input type="text" value="Aloitusjäte"/>
Kpl vaaka	Pöytävaaka
<input type="text" value="1"/>	<input type="text" value="3"/>
	Stopparivaaka
	<input type="text" value="4"/>

Tiivistepystykarmi

Pituus	Kiinnitys
<input type="text" value="2090"/>	<input type="text" value="kiinnitysreikä normaa"/>
Lukkoh	Kynnys
<input type="text" value="0"/>	<input type="text" value="ylös"/>
Tlukkoh	Ovityyppi
<input type="text" value="0"/>	<input type="text" value="yksittäinen ovi: oik"/>
Saranasiirto	Sarana
<input type="text" value="100"/>	<input type="text" value="FM23"/>
Alaväli	Ylivienti
<input type="text" value="31"/>	<input type="text" value="ei"/>
Korjaus	Murtotappi
<input type="text" value="12"/>	<input type="text" value="ei"/>
Turvalukko	MG Pysty
<input type="text" value="ei"/>	<input type="text" value="ei"/>
Jätevaaka	Telki
<input type="text" value="Aloitusjäte"/>	<input type="text" value="vakio"/>
Kpl pysty	Pöytäpysty
<input type="text" value="1"/>	<input type="text" value="3"/>
	Stopparipysty
	<input type="text" value="4"/>

Kuvio 8 Tiivistekarmit

4 PROJEKTIN VAIHEET

4.1 Aloitus

Projektin ensimmäinen vaihe oli asiakastapaaminen, jossa määriteltiin, mikä parametriohjelma on, mitä se tekee ja miksi tarvitaan uutta versiota. Se on ohjelma, joka luo työohjeen tuotantokoneille, parametritiedoston. Se kirjoittaa tuoteominaisuudet tiedostoksi työkoneelle, joka tekee siinä lueteltujen parametrien mukaisen tuotteen. Uudesta versiosta piti tehdä selkeämpi käyttää, sen pitää osata tehdä vanhan parametriohjelman asiat, sekä siihen pitää luoda toista konetta varten uusi käyttösiivu. (**Taulukko 1.**)

Taulukko 1 Vaatimusluettelo

Viite	Vaatimuksen kuvaus	Tärkeys
T1	Selkeä ulkoasu.	1
T2	Salvagninin S4-tiedostojen luominen.	1
T3	Pivaticin tiedostojen luominen.	1
T4	Dataa pitää pystyä muuttamaan.	1
1 = Pakollinen ominaisuus		

4.2 Suunnittelu

Ohjelmaa suunniteltaessa ohjelmistokieliä valikoituivat JavaScript, CSS, ja HTML, koska kehitysympäristö Electron käyttää perinteisiä Web-teknologioita. Myös Java, C# sekä aikaisemmassa versiossa käytetty Visual Basic oli suunnitteluvaiheessa mukana, mutta ne jäivät pois suunnittelun edetessä. Vaikka Java on käytetyin ohjelmistokieli tietoteknisissä sovelluksissa, se olisi ollut turhan raskas suorittaa huomioon ottaen millainen lopputulos työllä on. C# oli seuraava

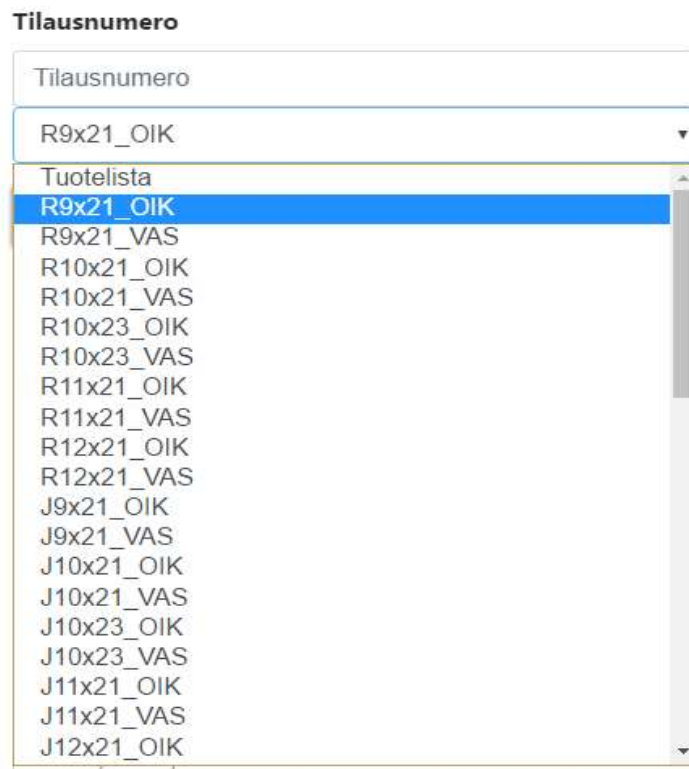
mahdollinen vaihtoehto, mutta versioita parametriohjelmasta olisi ollut monimutkaisempi testata. Visual Studiolla paketoitujen sovellukset vaativat .NET-puitteet asennettuna tietokoneelle toimiakseen. Jotta koko toiminnallisuutta pystyisi testaamaan, ohjelmasta pitää viedä testiversio asiakkaan tietokoneelle ja testata siellä paikallisesti. Ohjelmalla tehdään testaamista varten testitiedostot, josta tarkistetaan mahdolliset virheet. Tiedosto ladataan erilliseen ohjelmaan, joka valmistelee tiedostossa olevien parametrien pohjalta esikatselun, sekä ilmoittaa virheistä. Koska Visual Basic for Applications on periaatteeltaan melko vanhentunut kieli, se ei olisi ollut täysin kelvollinen vaatimusta ulkoasuun koskien. Koska ulkoasusta oli tarkoitus tehdä selkeä sekä moderni, JavaScript ja Electron päätyivät parhaaksi vaihtoehdoksi toteuttaa projekti.

4.3 Toteutus

Metallilevykoneet käyttävät niihin ohjelmoituja parametreja tunnistukseen minkälainen ovi pitää valmistaa, ja parametriojelma on apuväline parametritiedostojen luomiselle. Vanha ohjelma sisältää käyttöikkunan vain Salvagninille. Uusi ohjelma sisältää valikot sekä Salvagninille että Pivaticille. Salvagninin valikoissa on muistissa kokoelma vakiotuotteita, jotka voidaan valita tuotelistasta ja muokata tarvittaessa. Pivaticin valikot ovat suppeammat, mutta valikot löytyvät sekä karmeille että tiivistekarmeille. Molempien koneiden valikot jakavat metallilaatikon pituus- ja leveysmitat, tilausnumeron, sekä parioven käyntioven leveyden. Ohjelma kysyy ilmoituksella, jos uusi työohje yritetään kirjoittaa samalla tilausnumerolla ja ilmoittaa, kun tiedostot on luotu onnistuneesti.

Koska aikaisempi parametriojelma toimii Excel- ympäristössä makroilla, se pystyy hakemaan tuotetiedot Excel- taulukoista, joita voidaan muokata ja ohjelma päivittyy uudelleenkäynnistyksellä. Vastaavana keinona uusi ohjelma käyttää electron-json-storage- kirjastoa, joka tarkistaa onko tarvittavat tiedostot olemassa ja tarvittaessa luo tiedostot uudestaan. Uusi versio käyttää näin ollen tietojen säilyttämiseen JSON- tiedostoja, joita voidaan muokata lennossa ja ohjelma päivittää tiedot joko uudelleenkäynnistyksellä tai CTRL + R-

näppäinyhdistelmällä. Tiedot vakiotuotteisiin parsitaan käyttämällä IndexedDB:tä, joka on yksi Electroniin sisäänrakennetuista keinoista säilyttää ohjelmatietoja. Uudessa parametriohjelmassa on vanhan ohjelman tapaan pudotusvalikko Salvagninilla valmistettaville vakiovalinnoille, joka hakee tuotteen nimellä sitä vastaavat tiedot ja lataa ne käyttöikkunaan. (**Kuvio 9.**)



Kuvio 9 Vakio tuotelista

IndexedDB:hen on tallennettu vakioimitat teksti-ikkunoihin, kuin myös valintaruutujen totuusarvot, eli onko ruudussa valinta vai ei. (**Kuvio 10.**) Se myös valitsee pudotusvalikkoihin tuotteiden vakio-ominaisuudet. (**Kuvio 11.**)

Ikkunat

Käyntiovi <input checked="" type="checkbox"/>	Apuovi <input type="checkbox"/>
Reuna	Alareuna
119	999
Leveys	Leveys
538	-240
Yläreuna	Yläreuna
119	119
Korkeus	Korkeus
883	883

Kuvio 10 Teksti-ikkunat ja valintaruudut

Lukko

2590, 4190, ME115	
0	DF 3000
Abloy Exit <input type="checkbox"/>	Ei salpaa
Turvalukko <input type="checkbox"/>	DF 3000
0	6/3 salpa
Telkiaukko	Automaattisalpa
Ylivienti alas EA281 <input type="checkbox"/>	DF 330 reunasalpa
Ylivienti alas <input type="checkbox"/>	OLDA 26 HZ automaattisalpa
	AHFI 017
	AHFI 013
	B2390 PZA110(ota telki pois)
	Roca 200
	DF 3000+DF330 pohjaan
	DF 3000+DF330 kanteen

Kuvio 11 Pudotusvalikko

Kun kaikki tuotteen arvot on valittu ohjelmaan, käyttäjä painaa ”Lataus”-nappia, josta ohjelma luo parametritiedostot metallilevykoneelle ladattavaksi. (**Kuvio 12.**)



Kuvio 12 Tiedostojen luonti

Ohjelma lukee jokaisen arvon käyttöikkunassa, tekstiruudut, valintaikkunat ja pudotusvalikot ja luo niitä vastaavat parametrit parametritiedostoon.

Pivaticin parametritiedostot luodaan samalla tavalla, valitaan tarvittavat tiedot pudotusvalikoista, syötetään tekstiruutuihin oikeat mitat sekä valitaan joko tavalliset karmit tai tiivistekarmit. Ohjelmassa on käytetty jQuery-UI:ta havaitsemaan valinta, jonka perusteella kirjoitetaan parametritiedostot Pivaticille. Bootstrap ei täysin tukenut tätä ominaisuutta, joten piti keksiä toinen keino valinnan havaitsemiselle. (**Kuviot 7-8.**)

Viimeiseksi käyttäjä valitsee polun, jonne haluaa tiedostojen tallentuvan ja painaa ”Lataus”-nappia. (**Kuvio 15.**) Parametriohjelma luo parametritiedoston, windows-komentorivitiedoston sekä tekstitiedoston, joka kertoo tuotantokoneelle mikä tiedosto aloittaa työtehtävän Pivaticilla. (**Kuviot 13-14.**) Käyttöikkunassa on toinen tekstiruutu, johon kirjoitetaan polku, mihin tiedostot on tallennettu Pivaticin tietokoneella. Tuotteen valmistus aloitetaan suorittamalla windows-komentorivitiedosto, joka ensimmäiseksi poistaa edellisen työtehtävän, ”orderinput.dat”, jonka jälkeen se kopioi polussa sijaitsevan tiedoston ja tekee siitä uuden työtehtävän.


```
del C:\Flexpunch\orderinput.dat
copy D:\Tilaukset\4512_tilaus_r_valio_vaaka.txt C:\Flexpunch\orderinput.dat
```

Kuvio 13 Windows-komentorivitiedosto

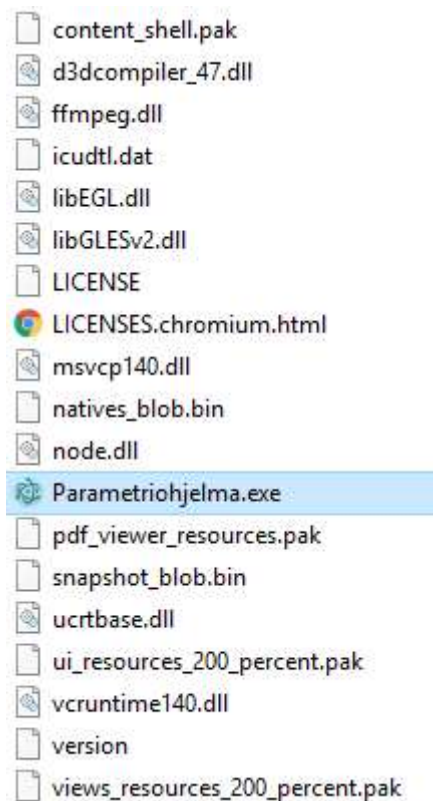
```
BEGIN punching
D:\Tilaukset\4512_Mitta_R_valio_vaaka.txt ;
END punching
```

Kuvio 14 Tekstitiedosto

Tilausnumero	Tiiv.Rainaleveys
<input type="text" value="Tilausnumero"/>	<input type="text" value="567"/>
Rainaleveys	
<input type="text" value="493"/>	
Vahvuuskorjaus	
<input type="text" value="2mm"/>	
Lämpö	
<input type="text" value="ei"/>	
Valitse polku:	<input type="text" value="C:\jokin\polku"/>
Lataus	<input type="text" value="D:\Tilaukset\"/>

Kuvio 15 Pivatic, lataus

Jokaisen kehityssyklin jälkeen kehitysovelluksesta luotiin testaussovellus, esim. tässä työssä käytetään ”Electron-Packager” -kirjastoa. Se paketoi lähdekoodin ja luo tarvittavat tiedostot käyttöä varten (**Kuvio 16.**). Sovellus ei myöskään tarvitse erinäistä asentamista, vaan sen voi käynnistää suoraan kansista löytyvästä suoritustiedostosta. (**Kuvio 16.**) Tämä mahdollisti vaivattoman testaamisen sovelluksella asiakkaan kanssa.



Kuvio 16 Sovelluskansio

4.4 Testaus

Versioita ohjelmasta testattiin tapaamisilla asiakkaan kanssa, jotka hoidettiin hänen toimistossaan yhtiön tuotantotiloissa. Niissä käytiin läpi mitä on tehty, muutokset edelliseen versioon, käytännön testaaminen ja mitä tehdään seuraavaan versioon. Käytännön testaaminen hoitui asiakkaan työkoneella, jonne purettiin kehitysversio ohjelmasta, ja luotiin testitiedostot parametreista. Tiedostot tarkistettiin virheiden varalta, joista oleellinen oli parametrien oikeellisuus. Tässä pitää tietää, että tuotantokoneiden parametritiedostot eroavat toisistaan, jonka takia Salvagninia varten on erillinen ohjelma. Salvagninia varten luodut tiedostot luodaan kansioon, josta valitaan toisella ohjelmalla suoritettava tiedosto. Se tarkistaa parametrit kääntämällä luodut tiedostot Salvagninin ohjelmalla, joka luo esikatselun tuotteesta ja ilmoittaa mahdollisista virheistä. Sen jälkeen ohjelmalla ladataan parametrit tuotantokoneelle tuotantokäskyksi, joka valmistaa tuotteen. Pivaticin parametrien testaamisessa oleellista oli, että uusi parametriojelma luo

parametritiedoston oikein käyttöikkunassa tehtyjen valintojen perusteella. Parametritiedoston lisäksi sen pitää myös luoda muut tiedostot.

```

begin punching
L=2090
B=493
A=0
I=12
D=0
C=0
U=0
O=0
F=2
J=0
M=0
N=0
E=1
H=0
Q=0
G=0
Z=100
W=31
V=1
P=3
S=4
R=1
D:\Pivatic_parametri\r_valio_karmi_parametri\alkuleikkaus B=B R=R ;
D:\Pivatic_parametri\r_valio_karmi_parametri\ohjelma L=L B=B C=C P=P S=S O=O V=V F=F J=J M=M N=N E=E H=H A=A Q=Q G=G ;
End punching

```

Kuvio 17 Pivatic, parametritiedosto

5 YHTEENVETO

Projektissa saatiin valmiiksi vaatimukset täyttävä sovellus, jossa on päivitetty käyttösiivu Salvagninille ja uusi käyttösiivu Pivaticille. (**Kuviot 18-19.**) Ohjelmalla pystytään luomaan parametritiedostot molemmille tuotantokoneille, ja tietopankkina käytettävää json-tiedostoa muokkaamalla voidaan ohjelman sisältämien tuotteiden ominaisuuksia ja parametriarvoja päivittää tarvittaessa.

Suurin syy miksi työ toteutettiin Electronilla, oli henkilökohtainen kiinnostus sekä JavaScript-taitojen kehittäminen. Työtä aloittaessa oli JavaScriptistä, web-teknologioista sekä Electron-alustasta tietämystä enimmäkseen perusasiat ja projektin edetessä tuli uutta tietoa ja osaamista niin itse kielestä kuin erilaisista web-ohjelmoinnissa käytettävistä teknologioista.

JavaScript jakaa monien sovelluskehittäjien mielipiteitä ohjelmointikielenä ja työpöytäsovelluksien kehittämiseen se on vähintäänkin epätavallinen. JavaScript kuitenkin toimii Electronin kanssa hyvin, sillä se on helppo ottaa käyttöön ja aloittaa suoraan sovelluksien kehittäminen. Sisäänrakennetun node.js:n ja Electronin itsensä ansiosta kehitysalusta tarjoaa laajan kirjon metodeja sekä rajapintoja, joita käyttää hyödyksi ilman kolmansien osapuolien kirjastojen asentamista.

LÄHDELUETTELO

Electron. Viitattu osoitteesta 19.9.2017 <https://electronjs.org/docs/tutorial/about>

Electron-userland. Viitattu osoitteesta 19.9.2017 <https://github.com/electron-userland/electron-packager>

Finder. Viitattu osoitteesta 19.9.2017
<https://www.finder.fi/Ovia+ja+ovien+valmistusta/Forssan+Metallity%C3%B6t+Oy/Forssa/yhteystiedot/134090>

Forssan Metallityöt. Viitattu osoitteesta 19.9.2017
<https://www.forssanmetallityot.fi/fi/>

Google. *Chromium*. Viitattu osoitteesta 19.9.2017
<https://www.chromium.org/Home>

Cruz Viotti, J. *GitHub Page electron-userland electron-json-storage*. Viitattu osoitteesta 19.9.2017 <https://github.com/electron-userland/electron-json-storage>

Mozilla. Viitattu osoitteesta 19.9.2017 https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB_API

Node.js Foundation. *Nodejs*. Viitattu osoitteesta 19.9.2017 <https://nodejs.org/en/>

Nokes, C. Viitattu osoitteesta 19.9.2017 <https://codeburst.io/deep-dive-into-electrons-main-and-renderer-processes-7a9599d5c9e2>

Twitter. *Twitter*. Viitattu osoitteesta 19.9.2017
<https://getbootstrap.com/docs/4.0/about/history/>

Wikipedia. Viitattu osoitteesta 19.9.2017 <https://fi.wikipedia.org/wiki/MIT-lisenssi>

LIITE 1

Tilausnumero		Käsisyys / Sarana		Ikkunat		POHJA		KANSI	
Tilausnumero		Oik FM23		Käyntiovi		DC01		DC01	
Tuotelista		Saranasiirto (<1403)		Reuna		1035		1035	
Laatikon leveys		100		Leveys		2145		2145	
Apuovi		Keskisarana pois, laatikko <1690		Yläreuna		1		1	
Laatikon leveys		4 Saranaa		Korkeus		1 mm		1 mm	
Apuovi		Murtotappi		Ikkunasiirto		RST / ALU		Korkeapaine	
Laatikon pituus		IV-reikä		Jäte kiinni		Trimmaus			
Karmimitta		150		Lista		Lämpökatko		100	
Karmimitta		126		Listankorkeus		P4: Kääntöpöytä		P4: Kääntöpöytä	
Laatikon paksuuskorjaus		Ovisilmä		Niitti listan		Kylkiluu kpl		Kylkiluu kpl	
Laatikon paksuuskorjaus		1500		Kannessa 20mm huullos		kylkiluu kpl		kylkiluu kpl	
YKSITTÄISET		Potkulevyt				P4: fm saranasiirto korke			
PARIOVET		Ladataan potkulevyt				Lataus		Aihio mitat	
TIIVISTEKARMI		Avauspuoli				Valitse kone			
KARMIYMPÄRI		Potkulevyt kansi				Salvagnini			
Lukko		Kannen korkeus							
Ei lukko työstöä		300							
Lukon korkeusmuutos		Pohjan korkeus							
DF 3000		300							
Abloy Exit		Sulkijapuoli							
Turvalukko		Potkulevyt pohja							
Turvalukon korkeusmuut		Paksuus							
Telkiaukko		0.8							
Ylivienti alas EA281		Ei vedintä							
Ylivienti alas APUOVI		Ylivienti keskelle EA281							
Mg-kosketin EA 502		Ylivienti keskelle APUOVI							
Lukijan merkkireikä kansi		Mg-kosketin 2xEA 502							
		Lukijan merkkireikä pohja							

Kuvio 18 Uusi parametrionhjelma, Salvagnini

LIITE 2

		Karmit		Tiivistet			
Tilausnumero <input type="text" value="Tilausnumero"/>		Tiiv.Rainaleveys <input type="text" value="567"/>					
Rainaleveys <input type="text" value="493"/>							
Vahvuuskorjaus <input type="text" value="2mm"/>							
Lämpö <input type="text" value="ei"/>							
Valitse polku: <input type="text" value="C:\Users\Teme1\Documents"/>							
Lataus <input type="text" value="D:\Tilaukset\"/>							
		Valio vaakakarmi		Valio pystyakarmi		Valitse kone Pivatic	
		Leveys <input type="text" value="990"/>	Kiinnitys <input type="text" value="kiinnitysreikä normaal"/>	Pituus <input type="text" value="2090"/>	Kiinnitys <input type="text" value="kiinnitysreikä normaal"/>		
		Korjaus <input type="text" value="12"/>	Ovityyppi <input type="text" value="yksittäinen ovi"/>	Lukkoh <input type="text" value="0"/>	Kynnys <input type="text" value="ei"/>		
		Käyntioven leveys <input type="text" value="0"/>	Kätisyys <input type="text" value="oik"/>	Tiukkoh <input type="text" value="0"/>	Ovityyppi <input type="text" value="yksittäinen ovi: oik"/>		
			Mg Vaaka <input type="text" value="ei"/>	Saranasiirto <input type="text" value="100"/>	Sarana <input type="text" value="FM23"/>		
			Salpa <input type="text" value="6/3, df3000, roca200"/>	Alavali <input type="text" value="31"/>	Ylivienti <input type="text" value="ei"/>		
			Jätevaaka <input type="text" value="Aloitusjäte"/>	Korjaus <input type="text" value="12"/>	Murtotappi <input type="text" value="ei"/>		
		Kpl vaaka <input type="text" value="1"/>	Pöytävaaka <input type="text" value="3"/>	Turvalukko <input type="text" value="ei"/>	MG Pysty <input type="text" value="ei"/>		
			Stopparivaaka <input type="text" value="4"/>	Jätevaaka <input type="text" value="Aloitusjäte"/>	Telki <input type="text" value="vakio"/>		
				Kpl pysty <input type="text" value="1"/>	Pöytäpysty <input type="text" value="3"/>	Stopparipysty <input type="text" value="4"/>	

Kuvio 19 Uusi parametriojelma, Pivatic